In the Claims

Please and amend Claims 1-2, 8-9, and 15-16 all as shown below. Applicant respectfully reserves the right to prosecute any originally presented claims in a continuing or future application.

1. (Currently Amended) A system that provides a generic user interface testing framework, comprising:

a computer including a computer readable medium, and a processor operating thereon;

a software application source code, stored on the computer readable medium, wherein the software application source code defines a software application under development, including a graphical user interface as part of the software application, and wherein the software application source code executes on the computer to display its graphical user interface;

one or more different software test tools that can be invoked by a user to perform testing operations on the graphical user interface that is displayed while the software application is running, wherein each of the one or more different software test tools understand their own tool-specific scripting language;

a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands that are abstractions independent of any tool-specific scripting language, wherein the test case input file can be reused as necessary for testing against a software application's graphical user interface in any of the different software test tools;

an interpretive engine that executes on the computer, and that includes a plurality of dynamically loaded libraries corresponding to the plurality of different software test tools, and including a library for each of the one or more different software test tools, wherein the interpretive engine receives the generic interface commands defined in the test case input file, determines which software test tool the user is currently using, loads required libraries to map the generic interface commands to corresponding tool-specific testing operations, uses the software test tool to perform the testing operations on the software application's graphical user interface including translating the generic interface commands to tool-specific commands, and reports to the user the success or failure of the testing operations ~~receives and translates generic interface commands from a tester~~;

~~a native library for mapping the generic interface commands to native language understood by a particular test software tool; and~~

an editor that allows the user to edit the test case input file ~~a tester to enter a number of test commands or directives~~, wherein the editor includes a rules-based wizard for assisting the ~~tester~~ user in generating different generic interface commands ~~test commands and directive scripts~~; and

wherein the rules-based wizard includes a plurality of user interface testing operations and wherein the rules-based wizard guides the user to pick from among the plurality of user interface testing operations to build the different generic interface commands.

~~wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.~~

2. (Currently Amended) The system of claim 1 wherein the system includes the test software tool stored locally on the ~~same~~ computer ~~or machine~~.

3. (Original) The system of claim 1 wherein the test software tool is stored at another computer or machine.

4. (Original) The system of claim 1 wherein the editor provides a graphical interface to allow the tester to enter said test commands.

5. (Original) The system of claim 1 wherein the editor communicates the test commands as a script of directives.

6. (Original) The system of claim 1 wherein the test commands can be created offline and subsequently communicated to the interpretive engine.

7. (Original) The system of claim 1 wherein the test software tool can be removed and replaced with another test software tool.

8. (Currently Amended) A method for providing a generic user interface testing framework on a computer, comprising the steps of:

executing a software application source code stored on a computer readable medium, wherein the software application source code defines a software application under development, including a graphical user interface as part of the software application, and wherein the software the software application source code executes to display its graphical user interface;

providing one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running;

allowing a tester to enter a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands, wherein the test case input file can be reused as necessary by a user for testing against a software application's graphical user interface in the same or a different software test tool; ~~number of generic test commands or directives via an editor or interface wherein the editor includes a rules-based wizard for assisting the tester in generating test commands and directive scripts; and~~

receiving, at an interpretive engine, generic interface commands from a tester

translating, using [[an]] the interpretive engine, the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool,

wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.

9. (Currently Amended) The method of claim 8 wherein ~~the system includes~~ the test software tool is stored locally on the ~~same~~ computer ~~or machine~~.

10. (Original) The method of claim 8 wherein the test software tool is stored at another computer or machine.

11. (Original) The method of claim 8 wherein the editor provides a graphical interface to allow the tester to enter said test commands.

12. (Original) The method of claim 8 wherein the editor communicates the test commands as a script of directives.

13. (Original) The method of claim 8 wherein the test commands can be created offline and subsequently communicated to the interpretive engine.

14. (Original) The method of claim 8 wherein the test software tool can be removed and replaced with another test software tool.

15. (Currently Amended) A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

executing a software application source code, wherein the software application source code defines a software application under development, including a graphical user interface as part of the software application, and wherein the software the software application source code executes to display its graphical user interface;

providing one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running;

allowing a tester to enter a number of ~~generic~~ test commands or directives via an editor or interface wherein the editor includes a rules-based wizard for assisting the tester in generating test commands and directive scripts; and

receiving, at an interpretive engine, generic interface commands from a tester

translating, using [[an]] the interpretive engine, the generic interface commands from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool,

wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.

16. (Currently Amended) The computer readable medium of claim 15 wherein ~~the system includes~~ the test software tool is stored locally on the ~~same~~ computer ~~or machine~~.

17. (Original) The computer readable medium of claim 15 wherein the test software tool is stored at another computer or machine.

18. (Original) The computer readable medium of claim 15 wherein the editor provides a graphical interface to allow the tester to enter said test commands.

19. (Original) The computer readable medium of claim 15 wherein the editor communicates the test commands as a script of directives.

20. (Original) The computer readable medium of claim 15 wherein the test commands can be created offline and subsequently communicated to the interpretive engine.

21. (Original) The computer readable medium of claim 15 wherein the test software tool can be removed and replaced with another test software tool.